



DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ RAHMANI
Prénom ▶ Mohammad
Adresse ▶ 31 Boulevard Saint Michel
72000 Le Mans

Titre professionnel

Développeur Web & Web-Mobile

MODALITE D'ACCES :

- ☐ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.
Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- une déclaration sur l'honneur à compléter et à signer ;
- des documents illustrant la pratique professionnelle du candidat (facultatif)
- des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.



<http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Développer la partie Front-End d'une application web ou web mobile en intégrant les recommandations de sécurité	p.	5
▶ Réalisation du site Créativ'Box	p.	5
▶ Réalisation du site Alain GOHIN	p.	9
Développer la partie Back-End d'une application web ou web mobile en intégrant les recommandations de sécurité	p.	12
▶ Réalisation du site Créativ'Box	p.	12
▶ Réalisation du site Alain GOHIN	p.	16
Titres, diplômes, CQP, attestations de formation <i>(facultatif)</i>	p.	19
Déclaration sur l'honneur	p.	20
Documents illustrant la pratique professionnelle <i>(facultatif)</i>	p.	21
Annexes <i>(Si le RC le prévoit)</i>	p.	22

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1

Développer la partie Front-End d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°1 ► Réalisation du site **Créativ'Box**

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Intro :

Créativ'Box est un site créé dans le cadre de ma formation de développeur web et web mobile. J'ai créé ce site pour que les créateurs puissent partager leurs activités et aussi afficher leurs profils, pour orienter les visiteurs vers leurs pages web ou leurs réseaux sociaux.

1- Maquetter une application (wireframing du site)

J'ai créé le sitemap pour connaître toutes les pages dont ce site aura besoin. Puis, j'ai créé les wireframes des pages et un UML qui contient le diagramme de cas d'utilisation et le diagramme de classe (voir les annexes).

Pour ce faire :

- Réalisation et création du sitemap.
- Décomposition du site en pages (wireframing).
- Création de la maquette de la version mobile et écran pc pour chaque page.
- Réalisation du diagramme de cas d'utilisation, représentant les fonctionnalités requises par les visiteurs, les membres et l'administrateur.
- Réalisation du diagramme de classe.

2- Réaliser une interface utilisateur web statique et adaptable

À l'aide de wireframes, de la charte graphique et TailwindCSS, j'ai créé les pages du site de manière statique. Lors de la conception, j'ai essayé de rendre le site compatible avec la plupart des différents appareils tels que les téléphones et les ordinateurs de bureau.

Pour les pages statiques, les langages HTML et CSS seront utilisés.

- **Page Accueil** : Une barre de navigation, une section Hero avec un titre, une description, deux boutons et trois images. Une section « catégories » pour afficher la liste des catégories, une section « images récentes » pour afficher les 4 dernières images publiées, une section « membres populaires » pour afficher le profil des 3 membres les plus populaires, une section « explorer » pour afficher 4 images au hasard, une section « inscription » pour inviter les visiteurs à s'inscrire sur le site et un Footer.
- **Page À propos** : Afficher la description du site et expliquer ses objectifs.

DOSSIER PROFESSIONNEL (DP)

- **Page Catégories** : Affiche toutes les catégories du site et les 4 dernières images publiées de chaque catégorie.
- **Page Cat/ :name** : Affiche toutes les images de chaque catégorie.
- **Page Galerie** : Affiche toutes les images récentes et une barre de recherche pour trouver des images par membre ou par catégorie.
- **Page Galerie/ :id** : Affiche l' image et ses détails et un formulaire pour écrire des commentaires.
- **Page Inscription** : Un formulaire pour s'inscrire sur le site.
- **Page Connexion** : Un formulaire pour se connecter sur son compte utilisateur.
- **Page Profil** : Page du membre avec la possibilité de modifier son profil (Avatar, nom, Facebook, Instagram, TikTok). Envoyer des images avec un titre, une description et une catégorie. Afficher ses images pour les modifier ou les supprimer.
- **Page Admin** : Espace d'administration du site (CRUD Image, Catégories, modifier/supprimer un membre, voir des messages).

J'ai utilisé les propriétés **Flex**, **Grid** et **Media Queries** de CSS pour que la mise en page de l'élément s'adapte et s'affiche correctement sur tous les appareils. Avec l'aide de **Flex** et **Grid**, nous pouvons gérer la position de chaque élément et la manière dont les éléments sont placés les uns à côté des autres. Avec l'aide de **Media Queries**, nous pouvons gérer le style de chaque élément selon les différentes tailles d'écran. Toutes les pages sont responsives et les capacités de Grid et Flex de TailwindCSS sont utilisées pour ça.

3- Développer une interface utilisateur web dynamique

Pour créer une interface dynamique, j'affiche sur la page d'accueil les dernières images publiées. J'ai également intégré une navbar dynamique avec deux possibilités : l'affichage de l'avatar du membre ou l'affichage de deux boutons pour s'inscrire ou se connecter. J'ai utilisé Handlebars pour faire du « templating ».

Pour les pages **admin** et **profil**, j'ai utilisé la librairie JQuery et JSDOM pour passer les données sur les modaux.

4- Interface utilisateur avec une solution de gestion de son contenu

Pour la gestion du contenu dans la page profil, j'ai créé un formulaire pour que le membre puisse ajouter, modifier ou supprimer ses informations (Nom, réseaux sociaux) et modifier sa photo de profil.

Pour la publication de ses images, j'ai créé un autre formulaire pour sélectionner une image, ajouter un titre et une description, sélectionner une catégorie.

Pour gérer ses publications, j'ai créé un emplacement pour que le membre puisse les consulter, les modifier ou les supprimer.

5- Interface administrateur avec une solution de gestion de contenu

J'ai créé une sidebar pour que l'administrateur puisse accéder aux différents contenus, par exemple : images, membres, etc. ...

Dans chaque page, il y a deux boutons pour chaque item pour que l'administrateur puisse faire différentes actions, par exemple : des modifications, des suppressions, répondre à des messages, etc. ...

2. Précisez les moyens utilisés :

Lors de la création de la partie Front-End du site, j'ai utilisé les outils et technologies suivants :

- **Mockflow**, **Whimsical** et **gitmind** pour créer le sitemap, les diagrammes et les wireframes.
- **GitHub** pour le suivi du projet.
- Framework **TailwindCSS** et **Flowbite** pour créer l'apparence du site et pour le responsive.
- Le logiciel **Visual Studio code**, **handlebars** .

3. Avec qui avez-vous travaillé ?

J'ai imaginé l'idée et l'architecture de ce site seul. J'ai reçu l'aide de mon formateur là où j'avais besoin de ses conseils. Ma conjointe m'a aidé à produire du contenu et à éditer et écrire des textes en français.

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ► Cliquez ici pour taper du texte.

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du : Cliquez ici au : Cliquez ici

5. Informations complémentaires (facultatif)

Pendant ce projet, j'ai pu enrichir mes connaissances dans le développement. Apprendre de nouveaux langages (UML, ...), la gestion de projet, la création de sitemap et wireframe. Cela m'a également beaucoup aidé à améliorer mon vocabulaire et mon orthographe lors de la rédaction des documents.

Activité-type 1

Développer la partie Front-End d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°2 ► Réalisation du site **Alain GOHIN**

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Intro :

Dans le cadre de ma formation de développeur web et web mobile, j'ai effectué un stage pour Mr Alain GOHIN, un artisan charpentier couvreur. Le but de la création du site était de donner plus de visibilité à son entreprise en présentant toutes ses activités, les photos de différents chantiers qu'il a réalisés et la possibilité pour les internautes de le contacter.

1. Mise en place de l'architecture

Après avoir créé la partie Back-end, j'ai débuté la création de la partie Front-end du site avec ReactJS. J'ai utilisé le module Axios pour envoyer et récupérer les données entre le Back-end et le Front-end.

2. Réaliser une interface utilisateur web statique et adaptable

À l'aide de Sitemap, j'ai créé les pages du site de manière statique. J'ai créé des composants et je les ai utilisés pour certaines pages, par exemple le formulaire de contact. Lors de la conception, j'ai essayé de rendre le site compatible avec la plupart des différents appareils tels que les téléphones et les ordinateurs de bureau. Pour réaliser le style des pages et les composants, j'ai utilisé CSS au format modular.

- **Page Accueil** : Une barre de navigation, une section Hero avec un titre et une image, une description, un bouton pour aller à la page Contact. Puis une section « À propos », une section « Nos activités » avec 3 cartes pour naviguer sur la page « Activités », une section « Travaux » avec les cartes qui affichent les images et les descriptions de chaque chantier réalisé par Mr GOHIN. Une section « Galerie » qui affiche les dernières photos publiées, une section « Contact » qui affiche un formulaire pour envoyer un message à Mr GOHIN puis pour terminer un Footer.
- **Page À propos** : Présente le profil professionnel de Mr GOHIN avec un bouton qui dirige les internautes sur la page des travaux réalisés.
- **Page Activités** : Affiche toutes les activités réalisées par Mr GOHIN.
- **Page Galerie** : Affiche toutes les photos publiées par Mr GOHIN.
- **Page Travaux** : Pour chaque chantier réalisé, on affiche une photo avec la description tel que le

type d'activité, la date et le lieu de la réalisation.

- **Page Inscription** : Un formulaire temporaire pour que Mr GOHIN puisse créer son compte administrateur.
- **Page Connexion** : Un formulaire pour se connecter sur son compte administrateur.
- **Page Contact** : Un formulaire pour que les internautes puissent envoyer directement un message à Mr GOHIN.
- **Page Admin** : Espace d'administration du site (CRUD Activités, Photos, voir et répondre aux messages).

Toutes les pages sont responsives. J'ai utilisé CSS modular pour attribuer à chaque élément un style spécifique.

3. Développer une interface utilisateur web dynamique

Pour créer une interface dynamique, j'ai utilisé les technologies (**State** et **Props**) qui sont intégrées dans ReactJS, pour créer du contenu dynamique sur toutes les pages et tous les composants. J'ai également utilisé **Axios** pour récupérer les données API et **redux** pour gérer des states.

4. Interface administrateur avec une solution de gestion de contenu

J'ai créé une sidebar pour que l'administrateur puisse accéder aux différents contenus : photos, activités, messages.

Dans chaque page, il y a un tableau qui affiche la liste des contenus. Chaque contenu à deux boutons pour modifier (ou répondre pour les messages) ou supprimer qui affiche un modal.

2. Précisez les moyens utilisés :

Lors de la création de la partie Front-End du site, j'ai utilisé les outils et technologies suivants :

- **Heroku** pour le suivi du projet et mise en production.
- **ReactJs** pour construire l'interface du site.
- **Redux** pour gérer les states.
- **Axios** pour récupérer et envoyer les données au back-end.
- Les modules **NPM**.

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec Mr GOHIN pour connaître ses besoins pour la réalisation du site. Il m'a envoyé les informations et les photos nécessaires pour afficher les contenus sur le site.

4. Contexte

Nom de l'entreprise, organisme ou association ► Entreprise Alain GOHIN

Chantier, atelier, service ►

Période d'exercice ► Du : 28/03/2022 au : 25/04/2022

5. Informations complémentaires (facultatif)

Pendant ce projet, j'ai pu enrichir mes connaissances dans le développement et apprendre de nouvelles technologies (ReactJS, redux, axios...).

Activité-type

2

Développer la partie Back-End d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 1 ► Réalisation du site **Créativ'Box**

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Introduction :

Le Back-End de mon site permettra de gérer les données tels que les images, leurs catégories et leurs commentaires. Il gèrera également les échanges entre les membres du site et l'administrateur.

1- Création d'une base de données

Le système de gestion de ma base de données (SGBD) a été réalisé avec MySQL, PhpMyAdmin, MySQL session et le terminal de commande MySQL.

J'ai créé un diagramme de classe, cela m'a aidé pour créer la base de données, les tables et les relations entre elles avec PhpMyAdmin. Une fois le projet terminé, je l'ai mis sur un serveur VPS et j'ai transféré les données de localhost vers ce serveur.

Sur le serveur VPS, j'ai accordé l'accès de tous les privilèges à un compte, pour accéder à la base de données.

Lors de la création du site, j'ai modifié plusieurs fois la base de données. Par exemple, j'ai ajouté ou supprimé certaines colonnes des tables.

Avec la création d'un mini code et à l'aide du module mysqldump, j'ai effectué un backup de ma base de données dans un fichier, ce qui m'a également permis de travailler à la fois au centre de formation et à mon domicile.

2- Développement des composant d'accès aux données

Pour l'accès aux données de ma base de données, j'ai mis en place des contrôleurs qui permettent de réaliser un CRUD :

- GET : Méthode pour récupérer des données.
- POST : Méthode pour envoyer des données.
- PUT : Méthode pour modifier des données.
- DELETE : Méthode pour supprimer des données.

Dans certains cas, j'ai utilisé des « Middleware ». Par exemple : permettre l'accès aux pages « Profil » ou « admin ».

Pour envoyer des images, j'ai utilisé le NPM package « Multer » comme un middleware. Pour stocker les images et les fichiers CSS, j'ai utilisé « Express.static » qui est une fonction middleware pour servir des fichiers statiques.

Pour rendre mon code plus modulaire, j'ai utilisé handlebar (moteur de templating) et l'architecture **MVC (Model – View – Controller)**. Avec eux, on peut développer indépendamment des couches d'application indépendantes et enfin se connecter entre ces couches.

Pour les sections de traitement du site, j'ai utilisé des instructions pour importer et exporter des modules dans Nodejs.

Quand je lance des requêtes, j'utilise « **POSTMAN** » pour les tester et les contrôler.

3- Développement de la partie Back-End

J'ai mis en place le Back-End avec l'outil **NPM**. Les modules utilisés sont :

- **bcryptjs** : Chiffrer (hasher) des mots de passe. Ce module sécurise les mots de passe des nouveaux membres lors de leur inscription et compare le mot de passe de connexion avec le mot de passe stocké et crypté sur la base de données.
- **dotenv** : Sécuriser certaines variables spécifiques (variables d'environnement), par exemple : configuration de connexion à la base de données.
- **expressJS** : Conserver des performances optimales et fournir une exécution rapide grâce à ce framework minimaliste. Il fournit également des middlewares et des définitions de routage (routing).
- **express-handlebars** : Pour parser la partie vue (view) de mon site.
- **express-mysql-session** : Pour store les sessions sur la base de données mysql.
- **express-session** : Créer et gérer un middleware de session.
- **fs (filesystem)** : Donner l'accès des fichiers sur le serveur à une liste de fonctions définies pour : créer, lire, écrire, copier, renommer et supprimer.
- **method-override** : Dans certains cas, on ne peut pas utiliser les méthodes, par exemple : « PUT » et « DELETE ». Avec ce module, on peut avoir cependant cette possibilité.
- **multer** : Envoyer les images sur le serveur.

- **mysql** : Se connecter sur la base de données en NodeJS.
- **nodemailer** : Pour envoyer des E-mails.
- **Docker** : Gérer et exécuter mon projet NodeJS sur VPS.
- **sharp** : Modifier des images, par exemple : les redimensionner.
- **mysqldump** : Pour faire une sauvegarde de la base de données.
- **Nginx** : Serveur web (Reverse Proxy).

4- Composants de gestion de contenu

Dans la page de l'administrateur, il est possible de gérer les contenus et les membres.

Pour **les contenus**, l'administrateur peut :

- consulter (GET)
 - ajouter (POST)
 - modifier (PUT)
 - supprimer (DELETE)
- } des images, des commentaires, des catégories, ...

Il peut également consulter et répondre aux messages qui lui sont envoyés depuis la page contact.

Pour **les membres**, l'administrateur peut :

- consulter (GET)
 - ajouter (POST)
 - modifier (PUT)
 - supprimer (DELETE)
- } chaque profil

2. Précisez les moyens utilisés :

- Docker
- GitHub
- NodeJs
- MySQL
- PhpMyAdmin

DOSSIER PROFESSIONNEL (DP)

- Visual Studio Code
- Les modules NPM :
 - bcryptjs
 - dotenv
 - expressJS
 - express-handlebars
 - express-mysql-session
 - express-session
 - fs
 - method-override
 - multer
 - mysql
 - nodemailer
 - sharp
 - mysqldump
 - nodemon
 - postcss
 - rand-token

3. Avec qui avez-vous travaillé ?

J'ai reçu l'aide de mon formateur là où j'avais besoin de ses conseils. Ma conjointe m'a aidé à produire du contenu et à éditer et écrire des textes en français.

4. Contexte

Nom de l'entreprise, organisme ou association ► Cliquez ici pour taper du texte.

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du : Cliquez ici au : Cliquez ici

5. Informations complémentaires (facultatif)

Activité-type

2

Développer la partie Back-End d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 2 ► Réalisation du site **Alain GOHIN**

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Introduction :

J'ai débuté la création du site avec la partie Back-end pour créer l'API. Elle permettra d'établir la connexion entre la base de données et le Front-end.

1- Création d'une base de données

Le système de gestion de ma base de données (SGBD) a été réalisé avec MongoDB, MongoDBCompass et mongoose. J'ai créé un compte sur le site de MongoDB pour créer la base de données.

2- Développement des composants d'accès aux données

Grâce au module mongoose, la partie Back-end est connectée à la base de données. Pour chaque contenu, j'ai créé le schéma, les routes et le controller. Pour chaque route de l'API, j'ai utilisé les méthodes HTTP pour accéder à la base de données.

- GET : Méthode pour récupérer des données.
- POST : Méthode pour envoyer des données.
- PUT : Méthode pour modifier des données.
- DELETE : Méthode pour supprimer des données.

Dans certains cas, j'ai utilisé des « Middleware ». Par exemple : publier des photos, supprimer des messages.

Pour envoyer des photos, j'ai utilisé le module « Multer » comme un middleware.

Pour stocker les photos, j'ai utilisé « Express.static » qui est une fonction middleware pour servir des fichiers statiques.

Pour contrôler les routes de l'API, j'ai utilisé « **POSTMAN** ».

3- Développement de la partie Back-End

J'ai mis en place le Back-End avec l'outil **NPM**. Les modules utilisés sont :

- **bcryptjs** : Chiffrer (hasher) des mots de passe. Ce module sécurise le mot de passe de l'administrateur.

- **dotenv** : Sécuriser certaines variables spécifiques (variables d'environnement), par exemple : configuration de connexion à la base de données.
- **expressJS** : Un framework pour créer la partie back-end du site et l'API.
- **jsonwebtoken** : Pour créer un système d'authentification entre la partie client et la partie serveur.
- **multer** : Envoyer les photos sur le serveur.
- **nodemailer** : Pour envoyer des E-mails.
- **sharp** : Modifier des photos, par exemple : les redimensionner.

4- Composants de gestion de contenu

Dans la page de l'administrateur, il est possible de gérer les contenus.

Pour **les contenus**, l'administrateur peut :

- consulter (GET)
 - ajouter (POST)
 - modifier (PUT)
 - supprimer (DELETE)
- } des photos, des activités, des messages

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

- **NodeJS, ExpressJS et MongoDB** (avec interface graphique **mongoDBcompass**) pour la partie back-end et la base de données.
- Les modules NPM :
 - bcryptjs
 - dotenv
 - expressJS
 - jsonwebtoken
 - multer
 - nodemailer
 - mongoose

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec Mr GOHIN pour connaître ses besoins pour la réalisation du site. Il m'a envoyé les informations et les photos nécessaires pour afficher les contenus sur le site.

4. Contexte

Nom de l'entreprise, organisme ou association ► Entreprise Alain GOHIN

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du : 28/03/2022 au : 25/04/2022

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] Mohammad RAHMANI ,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis
l'auteur(e) des réalisations jointes.

Fait à LE MANS le 27/04/2022

pour faire valoir ce que de droit.

Signature :

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)